CLAIMS:

1. A functional system comprising a set of functions (F) requiring access to a collective resource (RSRC), the system comprising an interface (INT) adapted to implement an access scheme (AS) characterized by a plurality of states (S) passed through in a predetermined manner, a state (S) forming a possibility of access of a given length and defining an order of priority in accordance with which a function (F) can access the collective resource (RSRC).

2. A functional system as claimed in Claim 1, characterized in that there is at least one state (S) which defines an order of priority (PS) for a sub-set of functions (SUB) only.

3. A functional system as claimed in Claim 2, characterized in that the interface is adapted to jump to the next state in the case that no function included in the sub-set of functions (SUB) has a pending request.

4. A method of managing a functional system comprising a set of functions (F) and a collective resource (RSRC) to which the functions (F) require access, characterized in that the method comprises the following step:

- passing through a plurality of states (S) in a predetermined manner, a state (S) forming a possibility of access of a given length and defining an order of priority (PS) in accordance with which a function (F) can access the collective resource (RSRC).

5. A data processing system comprising a set of processors (P) requiring access to a collective memory (MEM), the data processing system comprising a memory interface (INT) adapted to implement an access scheme (AS) characterized by a plurality of states (S) passed through in a predetermined manner, a state (S) forming a possibility of access of a given length and defining an order of priority (PS) in accordance with which a processor (P) can access the collective memory (MEM).

6. A "computer program" product for a functional system comprising a set of functions (F) and a collective resource (RSRC) to which the function (F) require access, characterized in that the "computer program" product comprises a set of instructions which, when loaded into such a functional system, causes the functional system to carry out the method claimed in Claim 4.